14th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS, Suceava, Romania, May 24-26, 2018

Multi-Agent Cognitive System for Optimal Solution Search

Ababii Victor, Sudacevschi Viorica, Munteanu Silvia

Informatics and Systems Engineering Department
Technical University of Moldova
Chisinau, Republic of Moldova
Victor.ababii@calc.utm.md,
Viorica.sudacevschi@calc.utm.md,
Silvia.munteanu@calc.utm.md

Bordian Dimitrie, Calugari Dmitri, Nistiriuc Ana, Dilevschi Sergiu

Computer Sciences, Electronics and Communications Doctoral Schools
Technical University of Moldova
Chisinau, Republic of Moldova
Dimitrie.bordian@scec.utm.md,
Dmitri.calugari@scec.utm.md

Abstract— **Solving optimization problems is a challenging issue in almost all engineering applications. In this paper we present the design of a multi-agent system with cognitive properties meant to find optimal solution according to the game theory. The multi-agent system consists of two subsystems that compete for maximum profit for their teams. Subsystems have no communication capabilities with each other. The status of an agent can only be identified based on its actions on the environment. In order to find an optimal solution we use a Nash Equilibrium concept of a non-cooperative game. The implementation of the system is done on a microcontroller. The evaluation of the Nash Equilibrium convergence rate was performed for 4 cases. The results are presented in the form of convergence charts.**

Keywords—multi-agent system; cognitive systems; optimal solutions, Nash Equilibrium.

I. INTRODUCTION

Multi-Agent Systems (MAS) are collaborative intelligent systems composed of a number of interacting computational entities capable of solving complex problems based on minimal or reduced data processing resources. These systems consist of a number of intelligent software or hardware agents, homogeneous or heterogeneous, which exchange information and knowledge, have the ability to co-operate, coordinate and negotiate their activities [1, 2, 3, 4].

MAS are becoming more and more popular in such areas as economy, technology, mathematics, computing, networking, artificial intelligence, robotics, collaborative decision support systems, data mining, social science, etc. A MAS is a group of autonomous, interacting agent sharing a common environment, which they perceive with sensors and upon which they act with actuators. From a structural point of view, MAS represents a subclass of distributed, parallel or concurrent processing systems [5, 6], where the agents have autonomy (their own interests and knowledge). In these systems synchronization and coordination are done at the common decision level [7].

A particular interest in the MAS development is based on the use of two types of agents: agents who are directing their activities to solve the issue positively (positive agents) and agents who are opposed to solve the problem positively (negative agents). The evolution of these systems is based on game theory, in particular, on the concept of Nash Equilibrium. [8,9]. Nash Equilibrium is considered one of the most important concepts of game theory, which attempts to determine mathematically and logically the actions that participants of a game should take to secure the best outcomes for themselves.

The ability and correctness of solving a problem for a multi-agent system is determined by its cognitive capabilities [10]. Cognitive systems have the ability to build strategies based on relevant hypotheses and contextual data, providing information and knowledge to the agents. Collective intelligence is the most important effect of sharing data and information. Over time, it has been found that cognitive systems are one of the most important steps in the long process of adapting to the evolution of information technology [11].

The article is structured as follows. The mathematical model of the optimal solution search in multi-agents systems is described in section 2. The description of the multi-agent system and cognitive agent synthesis is presented in detail in section 3. The example of an agent design on a micro-controller is shown in section 4. Convergence rate analysis for four cases is presented in section 5. A brief summary and an outlook on future work is offered in section 6.

II. MATHEMATICAL MODEL

Let's consider the process $\boldsymbol{S} \subset \boldsymbol{R}^N$ defined in an $\boldsymbol{N}$-dimensional space. The dynamics of the process $\boldsymbol{S}$ is

978-1-5386-1495-2/18/$31.00 ©2018 IEEE

determined by the set of functions $F = \{f_1, f_2, ..., f_K\}$, where $F : S \to R^N$. The state of the process S is determined by the state vector $X = \{x_1, x_2, ..., x_N\}$ [12, 13].

The set of functions F act on the process S according to the following system of equations (1):

$$\begin{cases} f_1(x_1, x_2, ..., x_N) = 0 \\ f_2(x_1, x_2, ..., x_N) = 0 \\ ... \\ f_K(x_1, x_2, ..., x_N) = 0 \end{cases} \tag{1}$$

Based on the model (1), can be defined the search criteria for optimal solutions $\min/\max$ (2):

$$\begin{cases} S = \sum_{j=1}^{K_{\min}} f_j^2(x_1, x_2, ..., x_N) \to S_{\min} \\ S = \sum_{i=1}^{K_{\max}} f_i^2(x_1, x_2, ..., x_N) \to S_{\max} \end{cases} \tag{2}$$

where: $S_{\min}$ - the minimal solution for the process S; $S_{\max}$ - the maximal solution for the process S; $K_{\min}$ - the number of functions that minimize the process S; $K_{\max}$ - the number of functions that maximize the process S, $K_{\min} \cap K_{\max} = \varnothing$.

III. COGNITIVE AGENTS SINTHESIS

To solve the model (2) the Muti-Agent System consisting of two subsets of agents is proposed (Figure 1): The set of ***Positive Agents*** which solve the condition $S \to S_{\min}$, and the set of ***Negative Agents*** which solve the condition $S \to S_{\max}$. Positive agents generate control signals $A_{\min} = \{y_j, \forall j = \overline{1, K_{\min}}\}$, while negative agents generate control signals $A_{\max} = \{y_i, \forall i = \overline{1, K_{\max}}\}$. The state of the process S is identified by reading state signals $ES = \{x_n, \forall n = \overline{1, N}\}$.

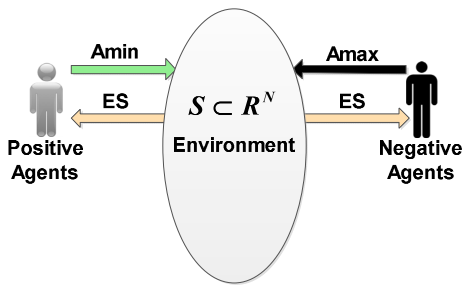


Fig. 1. Multi-Agent System

Each agent has information and knowledge that he assesses over time. The direct exchange of information between agents does not take place, the status of the positive and negative agents is determined by their actions carried out in the process. These criteria defined for agent behavior are part of the game theory and correspond to the Nash Equilibrium concept [9].

The behavioral model of the cognitive agent is presented in Figure 2, where: X_k - state vector of the process S; ADC - analog-to-digital converter; $RAM\ ES$ - state memory of the process; $\mathbf{Pr}$ - processor; $Rg\ Act$ - register that contains the binary code of the control signal applied to the process; DAC - digital-to-analog converter; y_k - analog control signal; $ROM\ Instr$ - program memory; $Cogn\ RAM$ - cognitive memory; $C[T-1]$ - cognitive model used to control the process at the time $T-1$; and $C[T]$ - the cognitive model calculated for the next step.

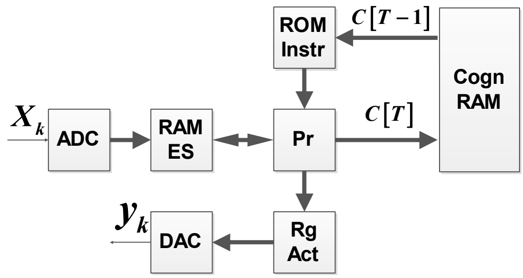


Fig. 2. Behavioral model of the cognitive agent

The agent's activity begins with the acquisition of the state vector X_k of the process S, converting it into digital code and store it in memory $RAM\ ES$. The processor $\mathbf{Pr}$ executes the code sequence from the program memory $ROM\ Instr$, which solves the model (1) with respect to the conditions (2). The code sequence executed by the processor $\mathbf{Pr}$ is determined by the content of the memory $Cogn\ RAM$, which changes its state at each iteration of the command as a result of the convergence rate analysis of the conditions (2). The binary results of the calculation necessary to control the process are written in the register $Rg\ Act$ and converted by DAC to analog control signal y_k.

Knowledge building stages are presented in Figure 3, where: X - the state of the process S; X/D - the conversion of the signal X to binary code ***Data*** acording to the knowledge model $C(X/D)$; D/I - conversion of the binary code ***Data*** to ***Information*** according to the knowledge model $C(D/I)$; I/C - conversion of the ***Information*** to ***Knowledge*** according to the knowledge model $C(I/C)$; I/D - conversion of the ***Information*** to ***Decision*** according to the knowledge model $C(I/D)$; y - control signal applied to the process S; $Knowledge[T-1]$ - knowledge models for conversion of data, information and knowledge; $Knowledge[T]$ - new knowledge, generated according to the previous information and knowledge. Knowledge represent methods and models used to process data. Data processing algorithms used by agents in their strategies are based on knowledge.

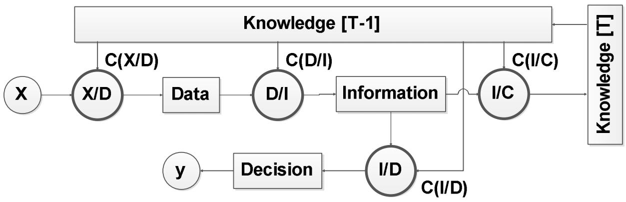


Fig. 3. Knowledge building stages

IV. Example of Agent Design

In order to validate the proposed method we have designed a version of an agent on ATmega328 micro-controller (Figure 4). Due to its specifications, ATmega328 allows to develop agents at a minimal cost and with good performance characteristics: 6 analog input pins are used to acquire the status signals X_k; 6 output Pulse-width modulation (PWM) pins are used to generate the control signal y_k; 32KB Flash memory with read-write capabilities is used for program code ($\boldsymbol{ROM\ Instr}$); 2 KB SRAM memory is used to store data ($\boldsymbol{RAM\ ES}$); 1KB EEPROM memory is used to store the accumulated knowledge ($\boldsymbol{Cogn\ RAM}$).

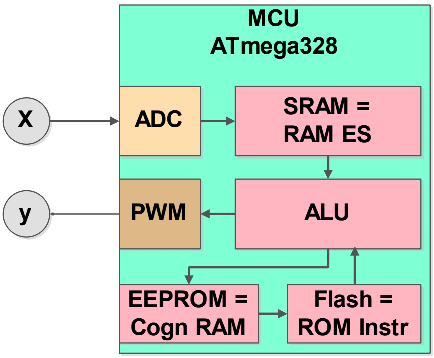


Fig. 4. Agent implementation on ATmega328 micro-controller

V. Convergence Rate Analysis

The convergence rate analyzing was performed for four cases. In each case we consider the different ratio of positive and negative agents.

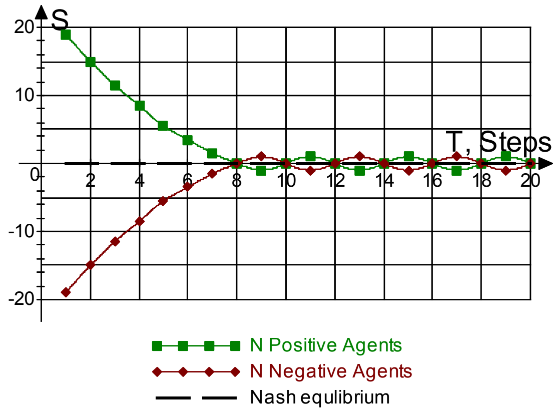
a) Figure 5,a. Multi-agent system consists of one positive agent and one negative agent. In this case, after solving the system of equations (1), with respect of the conditions (2), the Nash Equilibrium is established at step 15. After step 15 there is an oscillation of the balance determined by the fact that each agent try to obtain a higher profit as its opponent, but eventually this attempt is compensated by the opponent.

b) Figure 5,b. Multi-agent system consists of N positive agents and N negative agents. For the given case, the Nash Equilibrium is set at step 8. Reducing the number of steps to reach the Nash Equilibrium is determined by the presence of several agents in the system.

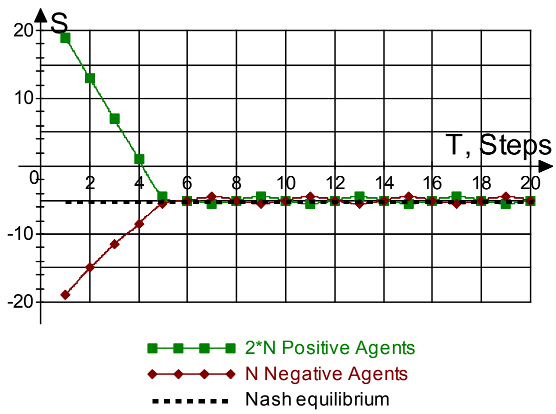
*c) Figure 5,c. Multi-agent system consists of 2*N positive agents and N negative agents.* In this case, the Nash Equilibrium is set at step 5, but the balance value is shifted in favor of the positive agents that are twice as many as negative ones.

*d) Figure 5,d. Multi-agent system consists of N positive agents and 2*N negative agents.* In this case, the Nash Equilibrium is set at step 5, but the balance value is shifted in favor of the negative agents that are twice as many as positive ones.

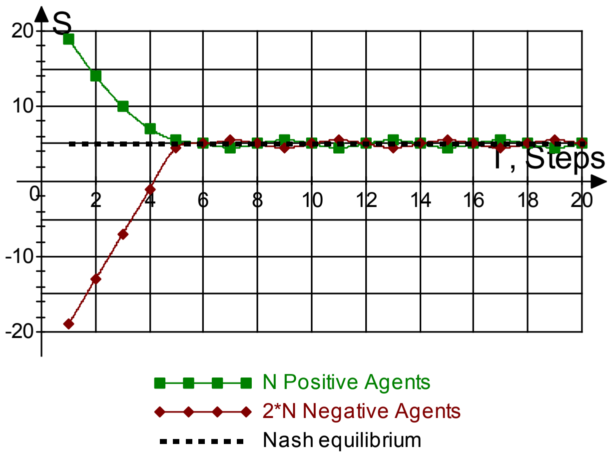
a) One Positive Agents and One Negative Agent.



b) N Positive Agents and N Negative Agents.



*c) 2*N Positive Agents N Negative Agents.*



*d) N Positive Agents and 2*N Negative Agents.*

Fig. 5. Convergence rate analysis

VI. Concluzion

The work described in this article applies the concept of Nash Equilibrium in order to find the optimal solution in a multi-agent system. Cognitive properties of agents allow the development of adaptive systems to the environment and infrastructure. The agents complexity is considerably reduced because they do not communicate with each other. The state and activity of the agents are determined only by their influence on the environment. Each agent solves its part of the global optimization problem. Positive agents use strategies that minimize the state parameters of the environment, negative agents use strategies that maximize the state parameters.

The evaluation of the Nash Equilibrium convergence rate was performed for 4 cases. In the case when the system consists of one positive agent and one negative agent, the Nash Equilibrium is established after a relatively large number of steps. For the system consisting of N positive and N negative agents, the convergence time was reduced twice. For the case when the system consists of 2 * N positive agents and N negative agents, the Nash Equilibrium is shifted in favor of positive agents. If more negative agents are used, the Nash Equilibrium is moved in their direction. The variation of the Nash Equilibrium for various cases of ratio between the number of positive and negative agents is determined by the fact that each agent solves its own optimization problem and influences the environment by minimizing or maximizing its state. The more agents are involved in the action on the environment, the faster and more pronounced this influence is.

Future research can take a number of directions: further experiments with more complex multi-agent systems with a larger number of agents; agents design with mixed strategies; development of non-linear action models on the work environment.

Acknowledgment

The results obtained in this paper are part of the research carried out under the theme "Models, methods and interfaces for the management and optimization of intelligent manufacturing systems" from the Department of Computer Science and Systems Engineering.

References

[1] M. Wooldridge, “An Introduction to MultiAgent Systems,” John Wiley & Sons. 2002, 366 p, ISBN: 0-471-49691-X.

[2] L. Rodriguez, E. Insfran and L. Cernuzzi, “Requirements Modeling for Multi-Agent Systems,” in Multi-Agent Systems - Modeling, Control, Programming, Simulations and Applications, Dr. Faisal Alkhateeb (Ed.), 2011, InTech, DOI: 10.5772/14776. (Available on: https://www.intechopen.com/books/multi-agent-systems-modeling-control-programming-simulations-and-applications/requirements-modeling-for-multi-agent-systems).

[3] M. Guijarro, R. Fuentes-Fernández and G. Pajares, “A Multi-Agent System Architecture for Sensor Networks,” in Multi-Agent Systems - Modeling, Control, Programming, Simulations and Applications, Dr. Faisal Alkhateeb (Ed.), 2011, InTech, DOI: 10.5772/14309. (Available on: https://www.intechopen.com/books/multi-agent-systems-modeling-control-programming-simulations-and-applications/a-multi-agent-system-architecture-for-sensor-networks).

[4] Y. Shoham, K. Leyton-Brown, “Multi-Agent systems. Algorithmic, Game-Theoretic, and Logiacal Foundations,” Published by Cambridge University Press. Revision 1.1, 2010. 532 p. (Available on: http://www.masfoundations.org).

[5] A. Hagit, J. Welch, “Distributed Computing: Fundamentals, Simulations, and Advanced Topics,” Wiley-Interscience, 2004, 432 p. ISBN: 0-471-45324-2.

[6] Gh. Safonov, V. Ababii, V. Sudacevschi, “Analysis of distributed computing architectures for synthesis of Multi-Agent systems,” European Applied Sciences Journal, № 9, 2016 (September), ISSN 2195-2183, pp. 34-37.

[7] G. Weiss, “Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence,” MIT Press, 1999, 615 p. ISBN: 0-262-23203-0.

[8] E. Rasmusen, “Games and Information: An Introduction to Game Theory,” 3rd ed. Blackwell, Oxford. 2001.

[9] F. Nash, “Non-Cooperative Games,” Annals of Mathematics 54, 286-295. 1951.

[10] H. I. Christiansen, A. Sloman, G.-J. Krujiff, J. Wyatt, “Cognitive systems,” Berlin: Springer-Verlag. 2009.

[11] D. Dumitrescu, “Algoritmi genetici şi strategii evolutive - Aplicaţii în inteligenţa artificială şi în domenii conexe,” Editura Albastră, Cluj-Napoca, 2000, 228p.

[12] A.R. Parkinson, R. Balling, and J.D. Hedengren, “Optimization Methods for Engineering Design,” Brigham Young University, 2013.

[13] I.C. Rădulescu, “Rezolvarea unor probleme de optimizare multi-obiectiv bazată pe algoritmi evolutivi,” Revista Română de Informatică şi Automatică, vol. 25. Nr. 2, 2015, pp.39-48.